

Method and System of Allocating Storage Resources in a Storage Area Network

Inventors: John Bates
Nicos Vekiarides

5

Cross-Reference to Other Applications

The following applications of common assignee are related to the present application, and are herein incorporated by reference in their entireties:

"Internet Protocol Data Mirroring," Ser. No. (to be assigned), Attorney Docket No. 1942.0040000, filed concurrently herewith.

10

"Method, System, and Computer Program Product for Managing Storage Resources," Ser. No. (to be assigned), Attorney Docket No. 1942.0050000, filed concurrently herewith.

15

"Method, System, and Computer Program Product for a Data Propagation Platform and Applications of Same," Ser. No. (to be assigned), Attorney Docket No. 1942.0070000, filed concurrently herewith.

Background of the Invention

Field of the Invention

The invention relates generally to the field of storage area networks, and more particularly to the allocation of storage in storage area networks.

20

Related Art

Traditional approaches exist for providing access to data in computer networks. These approaches generally fall into one of two categories: host-based and storage-based. Host-based approaches include those where storage management functionality is loaded and operated from the host (server).

Storage-based solutions are those where storage management functionality is loaded and operated from a storage array controller (or similar device).

Host-based approaches typically focus on application servers that run critical applications. For example, an application server may execute trading calculations for a trading room floor. Application servers are typically expensive, and are essential to a user's daily operations. Host-based storage solutions that run on application servers require processor cycles, and thus have a negative effect on the performance of the application server.

Additionally, host-based solutions suffer from difficulties in managing software and hardware interoperability in a multi-platform environment. Some of these difficulties include: managing separate licenses for each operating system; training system administrators on the various operating systems and host-based software; managing upgrades of operating systems; and managing inter-host dependencies when some functionality needs to be altered.

Storage-based solutions suffer from many of the same drawbacks. When storage-based solutions are provided with a disk array controller, compatibility between primary and target storage sites may become an issue. This compatibility problem may require a user to obtain hardware and software from the same provider or vendor. Moreover, hardware and software compatibility may also be limited to a particular range of versions provided by the vendor. Hence, if another vendor develops superior disk technology or connectivity solutions, a user may have difficulty introducing them into their existing environment.

Storage area networks (SANs) have been developed as a more recent approach to providing access to data in computer networks, to address some of the above concerns. A SAN is a network linking servers or workstations to storage devices. A SAN is intended to increase the pool of storage available to each server in the computer network, while reducing the data supply demand on servers. Conventional SANs, however, still may suffer from some of the above discussed problems, and some of their own.

For example, SANs may also suffer from problems associated with storage allocation. One problem relates to determining how to present the storage itself. For instance, it must be determined which storage devices shall be designated to provide storage for which servers. A further problem relates to storage security. It may be difficult for a SAN administrator to restrict access by certain servers to particular storage modules, while allowing other servers to access them. SAN administrators also have to confront the difficulty of coordinating networks that include a wide variety of different storage device types and manufactures, communication protocols, and other variations.

Therefore, in view of the above, what is needed is a system, method and computer program product for allocating storage in a storage area network. Furthermore, what is needed is a system, method and computer program product for allocating storage in a storage area network while maintaining storage security. Still further, what is needed is a system, method and computer program product for allocating storage from a variety of storage device types, manufactures, and interfaces in a storage area network.

Summary of the Invention

The present invention is directed to a system for allocating storage resources in a storage area network. A logical unit number (LUN) mapper receives at least one storage request parameter and maps the storage request parameters to at least one physical LUN. The LUN mapper includes at least one LUN map. The storage request parameters include a host id parameter, a target LUN parameter, and a target host bus adaptor (HBA) parameter. The LUN mapper uses the host id parameter to select the one of the LUN maps that corresponds to the host id parameter. The LUN mapper applies the target LUN parameter and the target HBA parameter to the selected LUN map to locate the physical LUN(s) stored in the selected LUN map. The LUN mapper issues the received read/write storage request to at least one storage device that houses the

physical LUN(s). The one or more storage devices are located in the storage area network.

5 In a further aspect of the present invention, a method for allocating storage in a storage area network is provided. A read/write storage request is received from a host computer. The read/write storage request is resolved. A physical LUN is determined from the resolved read/write storage request. A read/write storage request is issued to a storage device in a storage area network. The storage device corresponds to the determined physical LUN.

10 Further aspects of the present invention, and further features and benefits thereof, are described below. The accompanying drawings, which are incorporated herein and form a part of the specification, illustrate the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use the invention.

15 *Brief Description of the Figures*

In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit(s) of a reference number identifies the drawing in which the reference number first appears.

20 FIG. 1 illustrates a block diagram of an example storage allocator network configuration, according to embodiments of the present invention.

FIG. 2 illustrates a block diagram of a storage allocator, according to an exemplary embodiment of the present invention.

FIG. 3 illustrates an exemplary set of LUN maps, according to embodiments of the present invention.

25 FIG. 4 shows a flowchart providing detailed operational steps of an example embodiment of the present invention.

FIGS. 5-7 illustrate storage area network implementations of the present invention, according to embodiments of the present invention.

FIG. 8 illustrates an example data communication network, according to an embodiment of the present invention.

FIG. 9 shows a simplified five-layered communication model, based on an Open System Interconnection (OSI) reference model.

5 FIG. 10 shows an example of a computer system for implementing the present invention.

The present invention will now be described with reference to the accompanying drawings.

Detailed Description of the Preferred Embodiments

Overview

10 The present invention is directed to a method and system of allocating resources in a storage area network (SAN). The invention allocates storage resources in a SAN by mapping logical unit numbers (LUNs) representative of the storage resources to individual hosts, thereby allowing dynamic management of storage devices and hosts in the SAN. According to the present invention, each mapped LUN can be unique to a particular host or shared among different hosts.

15 FIG. 1 illustrates a block diagram of an example network configuration 100, according to embodiments of the present invention. Network configuration 100 comprises server(s) 102, a storage allocator 104, and storage 106. Generally, storage allocator 104 receives data I/O requests from server(s) 102, maps the data I/O requests to physical storage I/O requests, and forwards them to storage 106.

20 Server(s) 102 includes one or more hosts or servers that may be present in a data communication network. Server(s) 102 manage network resources. For instance, one or more of the servers of server(s) 102 may be file servers, network servers, application servers, database servers, or other types of server. Server(s) 25 102 may comprise single processor or multi-processor servers. Server(s) 102

process requests for data and applications from networked computers, workstations, and other peripheral devices otherwise known or described elsewhere herein. Server(s) 102 output requests to storage allocator 104 to write to, or read data from storage 106.

5 Storage allocator 104 receives storage read and write requests from server(s) 102 via first communications link 108. The storage read and write requests include references to one or more locations in a logical data space recognized by the requesting host. Storage allocator 104 parses the storage read and write requests by extracting various parameters that are included in the requests. In an embodiment, each storage read and write request from the host includes a host id, a target HBA (host bus adaptor), and a target LUN. Generally, a LUN corresponds to a label for a subunit of storage on a target storage device (virtual or actual), such as a disk drive. Storage allocator 104 uses the parsed read and write request to determine physical storage locations corresponding to the target locations in the logical data space. In a preferred embodiment, one or more LUN maps in storage allocator 104 are used to map the virtual data locations to physical locations in storage 106. Storage allocator 104 outputs read and write requests to physical storage/LUNs.

20 First communications link 108 may be an Ethernet link, a fibre channel link, a SCSI link, or other applicable type of communications link otherwise known or described elsewhere herein.

25 Storage 106 receives storage read and write requests from storage allocator 104 via second communications link 110. Storage 106 routes the received physical storage read and write requests to the corresponding storage device(s), which respond by reading or writing data as requested. Storage 106 comprises one or more storage devices that may be directly coupled to storage allocator 104, and/or may be interconnected in a storage area network configuration that is coupled to storage allocator 104. For example, storage 106 may comprise one or more of a variety of storage devices, including tape systems, JBODs (Just a Bunch Of Disks), floppy disk drives, optical disk drives, disk

arrays, and other applicable types of storage devices otherwise known or described elsewhere herein. Storage devices in storage 106 may be interconnected via SCSI and fibre channel links, and other types of links, in a variety of topologies. Example topologies for storage 106 are more fully described below.

Second communications link 110 may be an Ethernet link, fibre channel link, a SCSI link, or other applicable type of communications link otherwise known or described elsewhere herein.

According to the present invention, available storage is partitioned without any regard necessarily to the physical divisions of storage devices, and the partitions are stored in the LUN maps. These partitions are referred to as virtual or target LUNs. Portions of, or all of available physical storage may be partitioned and presented as virtual LUNs. Each host may be presented different portions of physical storage via the LUN maps, and/or some hosts may be presented with the same or overlapping portions. LUN maps allow the storage allocator of the present invention to make available a set of storage to a host, that may overlap or be completely independent from that made available to another host.

The virtual LUN configurations are stored in storage allocator 104 in LUN maps corresponding to each host. A LUN map may be chosen, and then used to convert virtual storage read or write requests by the respective host to an actual physical storage location. Embodiments for the storage allocator 104 of the present invention are described in further detail below.

Description in these terms is provided for convenience only. It is not intended that the invention be limited to application in this example environment. In fact, after reading the following description, it will become apparent to a person skilled in the relevant art how to implement the invention in alternative environments known now or developed in the future. Further detailed embodiments of the elements of network configuration 100 are discussed below.

Terminology related to the present invention is described in the following subsection. Next, an example storage area network environment is described, in which the present invention may be applied. Detailed embodiments of the storage allocator of the present invention are presented in the subsequent subsection, followed by exemplary storage area network implementations of the storage allocator. Finally, an exemplary computer system in which the present invention may be implemented is then described.

Terminology

To more clearly delineate the present invention, an effort is made throughout the specification to adhere to the following term definitions as consistently as possible.

- Arbitrated Loop** A shared 100MBps Fibre Channel transport supporting up to 126 devices and 1 fabric attachment.
- Fabric** One or more Fibre Channel switches in a networked topology.
- HBA** Host bus adapter; an interface between a server or workstation bus and a Fibre Channel network.
- Hub** In Fibre Channel, a wiring concentrator that collapses a loop topology into a physical star topology.
- Initiator** On a Fibre Channel network, typically a server or a workstation that initiates transactions to disk or tape targets.
- JBOD** Just a bunch of disks; typically configured as an Arbitrated Loop segment in a single chassis.
- LAN** Local area network; a network linking multiple devices in a single geographical location.

Logical Unit The entity within a target that executes I/O commands. For example, SCSI I/O commands are sent to a target and executed by a logical unit within that target. A SCSI physical disk typically has a single logical unit. Tape drives and array controllers may incorporate multiple logical units to which I/O commands can be addressed. Typically, each logical unit exported by an array controller corresponds to a virtual disk.

LUN Logical Unit Number; The identifier of a logical unit within a target, such as a SCSI identifier.

Point-to-point A dedicated Fibre Channel connection between two devices.

Private loop A free-standing Arbitrated Loop with no fabric attachment.

Public loop An Arbitrated Loop attached to a fabric switch.

RAID Redundant Array of Independent Disks.

SCSI Small Computer Systems Interface; both a protocol for transmitting large blocks of data and a parallel bus architecture.

SCSI-3 A SCSI standard that defines transmission of SCSI protocol over serial links.

Storage Any device used to store data; typically, magnetic disk media or tape.

Switch A device providing full bandwidth per port and high-speed routing of data via link-level addressing.

Target Typically a disk array or a tape subsystem on a Fibre Channel network.

Topology The physical or logical arrangement of devices in a networked configuration.

WAN Wide area network; a network linking geographically remote sites.

Example Storage Area Network Environment

In a preferred embodiment, the present invention is applicable to storage area networks. As discussed above, a storage area network (SAN) is a high-speed

sub-network of shared storage devices. A SAN operates to provide access to the shared storage devices for all servers on a local area network (LAN), wide area network (WAN), or other network coupled to the SAN. SAN attached storage (SAS) elements connect directly to the SAN, and provide file, database, block, or other types of data access services. SAS elements that provide such file access services are commonly called Network Attached Storage, or NAS devices. A SAN configuration potentially provides an entire pool of available storage to each network server, eliminating the conventional dedicated connection between server and disk. Furthermore, because a server's mass data storage requirements are fulfilled by the SAN, the server's processing power is largely conserved for the handling of applications rather than the handling of data requests.

FIG. 8 illustrates an example data communication network 800, according to an embodiment of the present invention. Network 800 includes a variety of devices which support communication between many different entities, including businesses, universities, individuals, government, and financial institutions. As shown in FIG. 8, a communication network, or combination of networks, interconnects the elements of network 800. Network 800 supports many different types of communication links implemented in a variety of architectures.

Network 800 may be considered to be an example of a storage area network that is applicable to the present invention. Network 800 comprises a pool of storage devices, including disk arrays 820, 822, 824, 828, 830, and 832. Network 800 provides access to this pool of storage devices to hosts/servers comprised by or coupled to network 800. Network 800 may be configured as point-to-point, arbitrated loop, or fabric topologies, or combinations thereof.

Network 800 comprises a switch 812. Switches, such as switch 812, typically filter and forward packets between LAN segments. Switch 812 may be an Ethernet switch, fast-Ethernet switch, or another type of switching device known to persons skilled in the relevant art(s). In other examples, switch 812 may be replaced by a router or a hub. A router generally moves data from one local segment to another, and to the telecommunications carrier, such as AT&T

or WorldCom, for remote sites. A hub is a common connection point for devices in a network. Suitable hubs include passive hubs, intelligent hubs, and switching hubs, and other hub types known to persons skilled in the relevant art(s).

5 Various types of terminal equipment and devices may interface with network 800. For example, a personal computer 802, a workstation 804, a printer 806, a laptop mobile device 808, and a handheld mobile device 810 interface with network 800 via switch 812. Further types of terminal equipment and devices that may interface with network 800 may include local area network (LAN) connections (e.g., other switches, routers, or hubs), personal computers with
10 modems, content servers of multi-media, audio, video, and other information, pocket organizers, Personal Data Assistants (PDAs), cellular phones, Wireless Application Protocol (WAP) phones, and set-top boxes. These and additional types of terminal equipment and devices, and ways to interface them with network 800, will be known by persons skilled in the relevant art(s) from the
15 teachings herein.

Network 800 includes one or more hosts or servers. For example, network 800 comprises server 814 and server 816. Servers 814 and 816 provide devices 802, 804, 806, 808, and 810 with network resources via switch 812. Servers 814 and 816 are typically computer systems that process end-user
20 requests for data and/or applications. In one example configuration, servers 814 and 816 provide redundant services. In another example configuration, server 814 and server 816 provide different services and thus share the processing load needed to serve the requirements of devices 802, 804, 806, 808, and 810. In further example configurations, one or both of servers 814 and 816 are connected
25 to the Internet, and thus server 814 and/or server 816 may provide Internet access to network 800. One or both of servers 814 and 816 may be Windows NT servers or UNIX servers, or other servers known to persons skilled in the relevant art(s).

30 A SAN appliance or device as described elsewhere herein may be inserted into network 800, according to embodiments of the present invention. For example, a SAN appliance 818 may to implemented to provide the required

connectivity between the storage device networking (disk arrays 820, 822, 824, 828, 830, and 832) and hosts and servers 814 and 816, and to provide the additional functionality of the storage allocator of the present invention described elsewhere herein.

5 Network 800 includes a hub 826. Hub 826 is connected to disk arrays 828, 830, and 832. Preferably, hub 826 is a fibre channel hub or other device used to allow access to data stored on connected storage devices, such as disk arrays 828, 830, and 832. Further fibre channel hubs may be cascaded with hub 826 to allow for expansion of the SAN, with additional storage devices, servers, and other devices. In an example configuration for network 800, hub 826 is an arbitrated loop hub. In such an example, disk arrays 828, 830, and 832 are organized in a ring or loop topology, which is collapsed into a physical star configuration by hub 826. Hub 826 allows the loop to circumvent a disabled or disconnected device while maintaining operation.

10 Network 800 may include one or more switches in addition to switch 812 that interface with storage devices. For example, a fibre channel switch or other high-speed device may be used to allow servers 814 and 816 access to data stored on connected storage devices, such as disk arrays 820, 822, and 824, via appliance 818. Fibre channel switches may be cascaded to allow for the expansion of the SAN, with additional storage devices, servers, and other devices.

15 Disk arrays 820, 822, 824, 828, 830, and 832 are storage devices providing data and application resources to servers 814 and 816 through appliance 818 and hub 826. As shown in FIG. 8, the storage of network 800 is principally accessed by servers 814 and 816 through appliance 818. The storage devices may be fibre channel-ready devices, or SCSI (Small Computer Systems Interface) compatible devices, for example. Fibre channel-to-SCSI bridges may be used to allow SCSI devices to interface with fibre channel hubs and switches, and other fibre channel-ready devices. One or more of disk arrays 820, 822, 824, 828, 830, and 832 may instead be alternative types of storage devices, including

tape systems, JBODs (Just a Bunch Of Disks), floppy disk drives, optical disk drives, and other related storage drive types.

The topology or architecture of network 800 will depend on the requirements of the particular application, and on the advantages offered by the chosen topology. One or more hubs 826, one or more switches, and/or one or more appliances 818 may be interconnected in any number of combinations to increase network capacity. Disk arrays 820, 822, 824, 828, 830, and 832, or fewer or more disk arrays as required, may be coupled to network 800 via these hubs 826, switches, and appliances 818.

Communication over a communication network, such as shown in network 800 of FIG. 8, is carried out through different layers. FIG. 9 shows a simplified five-layered communication model, based on Open System Interconnection (OSI) reference model. As shown in FIG. 9, this model includes an application layer 908, a transport layer 910, a network layer 920, a data link layer 930, and a physical layer 940. As would be apparent to persons skilled in the relevant art(s), any number of different layers and network protocols may be used as required by a particular application.

Application layer 908 provides functionality for the different tools and information services which are used to access information over the communications network. Example tools used to access information over a network include, but are not limited to Telnet log-in service 901, IRC chat 902, Web service 903, and SMTP (Simple Mail Transfer Protocol) electronic mail service 906. Web service 903 allows access to HTTP documents 904, and FTP (File Transfer Protocol) and Gopher files 905. Secure Socket Layer (SSL) is an optional protocol used to encrypt communications between a Web browser and Web server.

Transport layer 910 provides transmission control functionality using protocols, such as TCP, UDP, SPX, and others, that add information for acknowledgments that blocks of the file had been received.

Network layer 920 provides routing functionality by adding network addressing information using protocols such as IP, IPX, and others, that enable data transfer over the network.

5 Data link layer 930 provides information about the type of media on which the data was originated, such as Ethernet, token ring, or fiber distributed data interface (FDDI), and others.

Physical layer 940 provides encoding to place the data on the physical transport, such as twisted pair wire, copper wire, fiber optic cable, coaxial cable, and others.

10 Description of this example environment in these terms is provided for convenience only. It is not intended that the invention be limited to application in this example environment. In fact, after reading the description herein, it will become apparent to persons skilled in the relevant art(s) how to implement the invention in alternative environments. Further details on designing, configuring, and operating storage area networks are provided in Tom Clark, "Designing
15 Storage Area Networks: A Practical Reference for Implementing Fibre Channel SANs" (1999), which is incorporated herein by reference in its entirety.

Storage Allocator Embodiments

20 Structural implementations for the storage allocator of the present invention are described at a high-level and at a more detailed level. These structural implementations are described herein for illustrative purposes, and are not limiting. In particular, the present invention as described herein can be achieved using any number of structural implementations, including hardware, firmware, software, or any combination thereof. For instance, the present
25 invention as described herein may be implemented in a computer system, application-specific box, or other device. In an embodiment, the present invention may be implemented in a SAN appliance, which provides for an interface between host servers and storage. Such SAN appliances include the

SANLink™ appliance, developed by StorageApps Inc., located in Bridgewater, New Jersey.

Storage allocator 104 provides the capability to disassociate the logical representation of a disk (or other storage device) as presented to a host from the physical components that make up the logical disk. Specifically, the storage allocator of the present invention has the ability to change a LUN as presented to the server from the storage (a process also referred to as "LUN mapping," which involves mapping physical space to give a logical view of the storage). In an embodiment, LUN mapping works by assigning each host with a separate LUN map. For each incoming command, SCSI or otherwise, the system identifies the host, target host bus adapter (HBA), and target LUN. The system uses that information to convert the received virtual or target LUN to an actual or physical LUN, via a LUN map.

Through such LUN mapping, a host can be presented with any size storage pool that is required to meet changing needs. For example, multiple physical LUNs can be merged into a single storage image for a host (i.e., storage "expansion"). All connected storage units can be presented to a host as a single storage pool irrespective of storage area network topology. The result is that any storage subsystem (for example, SCSI or Fibre Channel) can be connected to any host (for example, UNIX or NT). Alternatively, physical LUNs can be partitioned into multiple images for a host (i.e., storage "partitioning").

Storage allocation through LUN mapping provides a number of desirable storage functions, including the following:

Partitioning: Through LUN mapping, a user may present a virtual device that is physically a partition of a larger device. Each partition appears to one or more hosts to be an actual physical device. These partitions may be used to share storage from a single disk (or other storage device) across multiple host operating systems.

Expansion: Through LUN mapping, a user may present a virtual device consisting of multiple merged physical LUNs, creating an image of an expanded

LUN to the outside world. Thus, for example, the user may consolidate both SCSI and Fibre Channel storage systems into the same global storage pool.

Security: The user may also manage access to storage via a LUN map "mask", which operates to limit the LUN(s) that a host sees. This masking may be used to prohibit a host from accessing data that it does not have permission to access.

In embodiments, the storage allocator of the present invention is based independently of a host. Unlike host-based solutions, which simply mask storage resources and require additional software on every host in the network, the present invention requires no additional software, hardware, or firmware on host machines. The present invention is supported by all platforms and all operating systems. Furthermore, host machines may be added to the network seamlessly, without disruption to the network.

FIG. 2 illustrates a block diagram of a storage allocator 104, according to an exemplary embodiment of the present invention. Storage allocator 104 comprises a network interface 202, a read/write storage request parser 204, a LUN mapper 206, and a SAN interface 208.

Network interface 202 receives a read/write storage request 210 via first communication link 108, shown in FIG. 1. Network interface 202 may include a host bus adaptor (HBA) that interfaces the internal bus architecture of storage allocator 104 with a fibre channel first communication link 108. Network interface 202 may additionally or alternatively include an Ethernet port when first communication link 108 comprises an Ethernet link. Further interfaces, as would be known to persons skilled in the relevant art(s), for network interface 202 are within the scope and spirit of the invention.

Read/write storage request parser 204 receives the read/write storage request 210 from network interface 202, extracts parameters from the read/write storage request 210, and supplies the parameters to LUN mapper 206. These parameters may include a host id, a target HBA, and a target LUN. The host id parameter includes the host id of the storage request initiator server. The target

HBA parameter is the particular HBA port address in storage allocator 104 that receives the read/write storage request 210. A server or host may be provided with more than one virtual storage view. The target LUN parameter is the logical unit number of the virtual storage unit to which the read/write storage request 210 is directed. In alternative embodiments, additional or different parameters may be supplied to LUN mapper 206.

LUN mapper 206 receives the extracted parameters from read/write storage request parser 204. LUN mapper 206 stores LUN maps corresponding to servers/hosts, as described above. As described above, available storage is partitioned in the LUN maps without any regard necessarily to the physical divisions of storage devices. These partitions are referred to as virtual or target LUNs. Portions of, or all of available physical storage may be partitioned and presented as virtual LUNs. Each host or server may be presented with different portions of physical storage via the LUN maps, or some hosts may have the same portions presented. Furthermore, each host or server has a set of labels or names it uses to refer to the virtual LUNs stored in the LUN maps, which may be the same as or different from another host's set of labels or names.

In an embodiment, LUN maps are identified by their respective server's host id value. Once identified, a LUN map may then be used to convert virtual storage read or write requests from its corresponding server or host to an actual physical storage location or LUN. FIG. 3 illustrates an exemplary set of LUN maps, according to embodiments of the present invention. FIG. 3 shows a first LUN map 302, a second LUN map 304, a third LUN map 306, and a fourth LUN map 308. While four LUN maps are shown in FIG. 3, the present invention is applicable to any number of LUN maps. First, second, third, and fourth LUN maps 302, 304, 306, and 308 are stored in LUN mapper 206.

As shown in FIG. 3, a LUN map may be a two-dimensional matrix. In a preferred embodiment, a LUN map stores a two-dimensional array of physical LUN data. A first axis of the LUN map is indexed by target LUN information, and a second axis of the LUN map is indexed by target HBA information.

In FIG. 2, SAN interface 208 receives a physical LUN read/write request from LUN mapper 206, and transmits it as physical LUN read/write request 212 on second communication link 110 (FIG. 1). In this manner, storage allocator 104 issues the received read/write storage request 210 to the actual storage device or devices comprising the determined physical LUN in the storage area network. SAN interface 208 may include one or more host bus adaptors (HBA) that interface the internal bus architecture of storage allocator 104 with second communication link 110. SAN interface 208 may support fibre channel and/or SCSI transmissions on second communication link 110. Further interfaces, as would be known to persons skilled in the relevant art(s), for SAN interface 208 are within the scope and spirit of the invention.

FIG. 4 shows a flowchart 400 providing operational steps of an example embodiment of the present invention. The steps of FIG. 4 may be implemented in hardware, firmware, software, or a combination thereof. For instance, the steps of FIG. 4 may be implemented by storage allocator 104. Furthermore, the steps of FIG. 4 do not necessarily have to occur in the order shown, as will be apparent to persons skilled in the relevant art(s) based on the teachings herein. Other structural embodiments will be apparent to persons skilled in the relevant art(s) based on the discussion contained herein. These steps are described in detail below.

The process begins with step 402. In step 402, a read/write storage request is received from a host computer. For instance, in embodiments, the read/write storage request may be received by network interface 202 described above.

In step 404, the read/write storage request is resolved. In embodiments, the parameters of host id, target LUN, and target HBA are extracted from the read/write storage request, as described above. In an embodiment, read/write storage request parser 204 may be used to execute step 404.

In step 406, one or more physical LUNs are determined from the resolved read/write storage request. In an embodiment, LUN mapper 206 may be used to execute step 406.

5 In step 408, an the read/write storage request is issued to one or more storage devices in a storage area network, wherein the storage devices correspond to the determined one or more physical LUNs. In an embodiment, the read/write storage request is a physical LUN read/write storage request 212. In an embodiment, the read/write storage request is issued by LUN mapper 206 to the storage area network via SAN interface 208.

10 In an exemplary embodiment of the present invention, step 406 may be implemented by the following procedure: To determine a physical LUN corresponding to a resolved read/write storage request, the proper LUN map must be selected. LUN mapper 206 uses the received host id parameter to determine which of the stored LUN maps to use. For example, LUN map 308 may be the proper LUN map corresponding to the received host id. As shown in FIG. 3, the received parameters, target LUN 310 and target HBA 312, are used as X- and Y- coordinates when searching the contents of a LUN map. Applying these parameters as X- and Y- coordinates to the determined LUN map 308, the corresponding physical LUN 314 may be located, and supplied as the actual physical storage location to the requesting server. As would be recognized by persons skilled in the relevant art(s) from the teachings herein, LUN maps may be organized and searched in alternative fashions.

20 The embodiments for the storage allocator of the present invention described above are provided for purposes of illustration. These embodiments are not intended to limit the invention. Alternate embodiments, differing slightly or substantially from those described herein, will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein.

Storage Area Network Embodiments of the Storage Allocator

The invention may be implemented in any communication network, including LANs, WANs, and the Internet. Preferably the invention is implemented in a storage area network, where many or all of the storage devices in the network are made available to the network's servers. As described above,
5 network configuration 100 of FIG. 1 shows a exemplary block diagram of a storage area network.

FIG. 5 illustrates a detailed block diagram of an exemplary storage area network implementation 500, according to an embodiment of the present invention. SAN implementation 500 comprises a first, second, and third host computer 502, 504, and 506, a first and second storage allocator 508 and 510, and a first, second, third, and fourth storage device 512, 514, 516, and 518.
10

Host computers 502, 504, and 506 are servers on a communications network. Storage allocators 508 and 510 are substantially equivalent to storage allocator 104. Storage allocators 508 and 510 parse storage read and write requests received from host computers 502, 504, and 506, and use the parsed read and write request to determine physical data locations in storage devices 512, 514, 516, and 518. Storage devices 512, 514, 516, and 518 are storage devices that receive physical storage read and write requests from storage allocators 508 and 510, and respond by reading data from or writing data to storage as requested.
15
20 SAN implementation 500 shows multiple storage allocators, storage allocators 508 and 510, operating in a single SAN. In further embodiments, any number of storage allocators may operate in a single SAN configuration.

As shown in SAN implementation 500, storage allocators 508 and 510 may receive read and write storage requests from one or more of the same host computers. Furthermore, storage allocators 508 and 510 may transmit physical read and write storage requests to one or more of the same storage devices. In particular, storage allocators 508 and 510 may transmit read and write storage
25

request to one or more of the same LUN partitions. Storage allocators 508 and 510 may assign the same or different names or labels to LUN partitions.

FIG. 6 illustrates a storage area network implementation 600, according to an exemplary embodiment of the present invention. SAN implementation 600 comprises storage allocator 104 and storage 106. Storage 106 comprises a loop hub 604, a first, second, and third disks 606, 608, and 610, a fibre channel disk array 612, a tape subsystem 616, and a SCSI disk array 618.

Storage allocator 104 may receive read and write storage requests from one or more of the host computers of server(s) 102. Furthermore, storage allocator 104 may transmit physical read and write storage requests to one or more of the storage devices of storage 106.

Tape subsystem 616 receives and sends data according to SCSI protocols from/to storage allocator 104.

SCSI disk array 618 is coupled to a SCSI port of storage allocator 104, for data transfer.

Loop hub 604 forms an arbitrated loop with first, second, and third disks 606, 608, and 610. Loop hub 604 and fibre channel disk array 612 are coupled to fibre channel ports of storage allocator 104 via fibre channel communication links.

As described above, multiple LUNs of the storage devices of storage 106 can be merged into a single image for a host (expansion) in server(s) 102. All connected storage units, including first, second, and third disks 606, 608, and 610, fibre channel disk array 612, SCSI disk array 618, and tape subsystem 616, can be presented to a host as a single storage pool irrespective of the storage device types or storage area network topology. As shown in FIG. 106, any storage subsystem types, including different types such as fibre channel disk array 612 and SCSI disk array 618, may be connected to any host in a single storage pool. Alternatively, the storage subsystems can be can be partitioned into multiple LUN portions for a host.

Furthermore, fewer than all storage devices of storage 106 may be made available to a server or host for purposes of security. For example, access to certain storage devices/physical LUNs may be limited to prevent corruption of data. Access to storage devices and/or physical LUNs may be managed by storage allocator 104 via one or more LUN maps, which may operate to limit the LUN(s) that a host sees. This limiting or masking may be used to prohibit a host from accessing data that it does not have permission to access. For example, all of, or any portion of first, second, and third disks 606, 608, and 610, fibre channel disk array 612, SCSI disk array 618, and tape subsystem 616 may be hidden from view to selected servers of server(s) 102 by storage allocator 104.

FIG. 7 illustrates a storage area network implementation 700, according to an exemplary embodiment of the present invention. SAN implementation 700 comprises server(s) 102, storage allocator 104, and storage 106. Server(s) 102 comprises an NT server 702, a first UNIX server 704, a second UNIX server 706, and a switch 708. Storage 106 comprises a fibre channel switch 710, a fibre channel disk array 712, and a SCSI disk array 714. SAN implementation 700 will be used to demonstrate an example of LUN mapping, as follows.

In this LUN mapping example, fibre channel disk array 712 includes two LUN divisions, with the following attributes:

Label	No. of spindles	Capacity
LUN 0	9	153 GBytes
LUN 1	5	72 GBytes

SCSI disk array 714 includes three LUN divisions, with the following attributes:

Label	No. of spindles	Capacity
LUN 0	2	9 GBytes
LUN 1	2	9 GBytes
LUN 2	5	36 GBytes

Therefore, according to the present example, storage allocator 104 has access to a total storage capacity of five LUNs, with 279 GBytes total, in SAN implementation 700.

In the present example, storage allocator 104 is configured to map or mask the total available storage capacity of 279 GBytes, and present it to the hosts in the form of eleven LUNs, with 181 GBytes total. 98 GBytes of the total available storage is not being made accessible to the hosts.

SAN implementation 700 provides for storage partitioning. Eleven LUNs, or virtual devices, that are a partition five physical LUNs. Each of the eleven LUNs appear to a host as an actual physical device. These virtual LUN partitions are used to share storage from the five physical LUNs across multiple host operating systems. In alternative embodiments, the five physical LUNs may be combined in a variety of ways to provide for storage expansion. In other words, the five LUNs may be merged to create an image of one or more expanded LUNs to the outside world.

SAN implementation 700 also provides for storage security. For example, the 98 GBytes of storage not provided to the servers may be designated as secure storage to be protected from access by the hosts. Furthermore, each server has access to its own set of LUNs. For example, this arrangement may aid in preventing data corruption.

In the present example, the available storage may be apportioned to NT server 702, first UNIX server 704, and second UNIX server 706 as follows.

NT server 702 is presented with three virtual LUNs in the following storage configuration:

Label	Capacity
LUN 0	9 GBytes
LUN 1	24 GBytes
LUN 2	24 GBytes

UNIX servers 704 and 706 are each presented with four virtual LUNS in the following storage configuration:

Label	Capacity
LUN 0	12 GBytes
LUN 1	30 GBytes
LUN 2	10 Gbytes
LUN 3	10 GBytes

In the present example, each host has access to separate, non-overlapping physical LUNs or storage locations. As shown, the hosts are provided with different LUNs having the same virtual LUN names. In alternative embodiments, the available storage may be partitioned and presented to the hosts in any number of different ways. For example, two or more hosts may have access to the same storage locations, and/or same LUNs.

In an embodiment, storage allocator 104 is implemented in a SAN appliance or other device. For instance, the SAN appliance/storage allocator may be implemented with aspects of a computer system such as described in further detail below. Users may have access to the computer system's graphical user interface (GUI) where they can manually configure the allocation of storage by the storage allocator. In embodiments, users may also configure operation of the storage allocator over a network, such as the Internet, via a GUI such as a web browser. For instance, the user may be able to designate certain storage devices and/or locations as secure areas, and then allocate storage to the hosts from the remaining locations of storage.

In alternative embodiments, the storage allocator may be configured to operate in a semi-automated or fully automated fashion, where storage is allocated by the operation of a computer algorithm. For instance, a user may be able to designate certain storage devices and/or locations as secure storage areas, or these may be designated automatically. Further, the computer algorithm may

then allocate the remaining storage by proceeding from top to bottom through the list of remaining storage devices and/or locations, allocating all of the remaining storage in this order until no further storage is required. Alternatively, storage may be allocated to match storage device access speed with the required storage access speed of specific hosts and applications. Further ways of allocating storage will be known to persons skilled in the relevant art(s) from the teachings herein.

It will be known to persons skilled in the relevant art(s) from the teachings herein that the invention is adaptable to additional or fewer servers, additional or fewer storage devices, and amounts of storage capacity different than presented in the example of SAN implementation 700, and the other SAN implementation examples. Description in these terms is provided for convenience only. It is not intended that the invention be limited to application in this example environment. In fact, after reading the following description, it will become apparent to persons skilled in the relevant arts how to implement the invention in alternative environments known now or developed in the future.

Example Computer System

An example of a computer system 1040 is shown in FIG. 10. The computer system 1040 represents any single or multi-processor computer. In conjunction, single-threaded and multi-threaded applications can be used. Unified or distributed memory systems can be used. Computer system 1040, or portions thereof, may be used to implement the present invention. For example, the storage allocator of the present invention may comprise software running on a computer system such as computer system 1040.

In one example, the storage allocator of the present invention is implemented in a multi-platform (platform independent) programming language such as JAVA 1.1, programming language/structured query language (PL/SQL), hyper-text mark-up language (HTML), practical extraction report language

(PERL), common gateway interface/structured query language (CGI/SQL) or the like. Java™- enabled and JavaScript™- enabled browsers are used, such as, Netscape™, HotJava™, and Microsoft™ Explorer™ browsers. Active content Web pages can be used. Such active content Web pages can include Java™ applets or ActiveX™ controls, or any other active content technology developed now or in the future. The present invention, however, is not intended to be limited to Java™, JavaScript™, or their enabled browsers, and can be implemented in any programming language and browser, developed now or in the future, as would be apparent to a person skilled in the art given this description.

In another example, the storage allocator of the present invention, including read/write storage request parser 204 and LUN mapper 206, may be implemented using a high-level programming language (e.g., C++) and applications written for the Microsoft Windows™ environment. It will be apparent to persons skilled in the relevant art(s) how to implement the invention in alternative embodiments from the teachings herein.

Computer system 1040 includes one or more processors, such as processor 1044. One or more processors 1044 can execute software implementing routines described above, such as shown in flowchart 400. Each processor 1044 is connected to a communication infrastructure 1042 (e.g., a communications bus, cross-bar, or network). Various software embodiments are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled in the relevant art how to implement the invention using other computer systems and/or computer architectures.

Computer system 1040 can include a display interface 1002 that forwards graphics, text, and other data from the communication infrastructure 1042 (or from a frame buffer not shown) for display on the display unit 1030.

Computer system 1040 also includes a main memory 1046, preferably random access memory (RAM), and can also include a secondary memory 1048. The secondary memory 1048 can include, for example, a hard disk drive 1050 and/or a removable storage drive 1052, representing a floppy disk drive, a

magnetic tape drive, an optical disk drive, etc. The removable storage drive 1052 reads from and/or writes to a removable storage unit 1054 in a well known manner. Removable storage unit 1054 represents a floppy disk, magnetic tape, optical disk, etc., which is read by and written to by removable storage drive
5 1052. As will be appreciated, the removable storage unit 1054 includes a computer usable storage medium having stored therein computer software and/or data.

In alternative embodiments, secondary memory 1048 may include other similar means for allowing computer programs or other instructions to be loaded
10 into computer system 1040. Such means can include, for example, a removable storage unit 1062 and an interface 1060. Examples can include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 1062 and interfaces 1060 which allow software
15 and data to be transferred from the removable storage unit 1062 to computer system 1040.

Computer system 1040 can also include a communications interface 1064. Communications interface 1064 allows software and data to be transferred between computer system 1040 and external devices via communications path
20 1066. Examples of communications interface 1064 can include a modem, a network interface (such as Ethernet card), a communications port, interfaces described above, etc. Software and data transferred via communications interface 1064 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communications interface 1064, via
25 communications path 1066. Note that communications interface 1064 provides a means by which computer system 1040 can interface to a network such as the Internet.

The present invention can be implemented using software running (that is, executing) in an environment similar to that described above with respect to
30 FIG. 8. In this document, the term "computer program product" is used to

generally refer to removable storage unit 1054, a hard disk installed in hard disk drive 1050, or a carrier wave carrying software over a communication path 1066 (wireless link or cable) to communication interface 1064. A computer useable medium can include magnetic media, optical media, or other recordable media, or media that transmits a carrier wave or other signal. These computer program products are means for providing software to computer system 1040.

Computer programs (also called computer control logic) are stored in main memory 1046 and/or secondary memory 1048. Computer programs can also be received via communications interface 1064. Such computer programs, when executed, enable the computer system 1040 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 1044 to perform features of the present invention. Accordingly, such computer programs represent controllers of the computer system 1040.

The present invention can be implemented as control logic in software, firmware, hardware or any combination thereof. In an embodiment where the invention is implemented using software, the software may be stored in a computer program product and loaded into computer system 1040 using removable storage drive 1052, hard disk drive 1050, or interface 1060. Alternatively, the computer program product may be downloaded to computer system 1040 over communications path 1066. The control logic (software), when executed by the one or more processors 1044, causes the processor(s) 1044 to perform functions of the invention as described herein.

In another embodiment, the invention is implemented primarily in firmware and/or hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of a hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s) from the teachings herein.

5

SKGF Ref. No. 1942.0030000